UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/682,330	10/09/2003	Sanjay P. Ghatare	21756-014500	9815

51206 7590 09/11/2007

TOWNSEND AND TOWNSEND AND CREW LLP
TWO EMBARCADERO CENTER
8TH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER	
SYED, FARHAN M	

ART UNIT	PAPER NUMBER
2165	

MAIL DATE	DELIVERY MODE
09/11/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/682,330	GHATARE, SANJAY P.	
	Examiner	Art Unit	
	Farhan M. Syed	2165	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 June 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,7 and 9-46* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,7 and 9-46* is/are rejected.

7)☒ Claim(s) *20* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *3/30/07*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-5,7, and 9-46 are pending.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claim1-5, 7, 9-14, 17, 21, 25, 26, 28, 30-35, 39, 44, and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burroughs et al (U.S. Patent No. 6,341,289 and known hereinafter as Burroughs) in view of Shiigi et al (U.S. Patent Pub. 2003/0014442 A1 and known hereinafter as Shiigi).

As per claims 1, 21, 25 and 28, Burroughs teaches a method for partitioning, comprising (i.e. *"The preferred embodiment of the present invention provides a system and method is provided that allows transparent and flexible partitioning of created objects in the context of schema mapping."* The preceding text clearly indicates that partitioning is flexible partitioning of created objects.)(Abstract): receiving a data access request, said data access request includes one or more variables in a first format (i.e. *"The first step 802 of method 800 is when an application makes a run time request to create a new owned entity for a specified owning entity."* The preceding text clearly indicates that a data access request is a run time request to create a new owned entity and one or more variable in the first format is inherent in the data object.)(Column 13, lines 32-35); determining one or more data stores of a plurality of data stores to service said data access request,

said step of determining includes accessing one or more mappings of said one or more variables to said plurality of data stores and using said mappings to evaluate partition expressions for said data stores (i.e. *"The schema mapping code defines the mapping between persistent storage in relational database tables and objects. As discussed above, this mapping preferably includes the defining and mapping of both partition key values and partition keys used to partition selected tables into two or more partitions. This mapping also defines the mapping of EntityOwningExtent (EOE) objects for each partition of each table."* The preceding text clearly indicates that one or more data stores of a set of data stores are inherent in a relational database tables and objects, one or more variables is the entityowningextent (EOE) objects.)(Column 13, lines 22-29); and sending information for said data access request to said one or more data stores determined to service said data access request (i.e. *"The next step 806 is to store the entity data for the owning entity in the appropriate row of the appropriate table if the data for the owning entity it has not been previous stored"* The preceding text clearly indicates that the storing the entity data for the owning entity in the appropriate row of the appropriate table is sending information for the said data access request to one or more data stores to service the request.)(Column 13, lines 42-44).

Burroughs does not explicitly teach a method wherein said plurality of data stores comprises at least one relational database and at least one Lightweight Directory Access Protocol (LDAP) directory.

Shiigi teaches a method wherein said plurality of data stores comprises at least one relational database and at least one Lightweight Directory Access Protocol directory (i.e. *"A Repository Manager 16A translates the logical design used in the system into the structure and format of the repository. A Resource Manager 18A* ***stores and manages resources in a Resource Repository*** *54. A repository is typically, but does not have to be, a relational database." "In this example, the Core System is implemented as a Java application that can run on the following application platforms for the various functions required in the system: Operating System; Java Application Server;* ***Relational***

***Database; and LDAP Directory Server**. At the bottommost layer, the Core System runs inside of a readily available operating system, such as Microsoft WINDOWS.TM. or freely available GNU/Linux on a server computer. The application code is written in the Java programming language and runs as a Web Application inside of a Web Application Server implementing the J2EE and Java Servlet 2.3 standards. Alternatively, a free application server, `Tomcat` is available from the Jakarta project of the Apache Foundation, while commercial application servers are sold by IBM, BEA Systems, and Macromedia Inc. among others. The Web Application Server is run inside of a Java Virtual-Machine (JVM) application that sits on top of the underlying computer operating system. JVM implementations are freely available from Sun Microsystems and IBM."* The preceding text clearly indicates that at least one relational database and an LDAP directory exists.)(Paragraphs [0040], [0088]).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Burroughs with the teachings of Shiigi to include a method wherein said plurality of data stores comprises at least one relational database and at least one Lightweight Directory Access Protocol directory with the motivation to integrate relational database systems with object oriented systems that provides for efficient partitioning of objects in the object oriented system. (Burroughs, column 3, lines 30-33).

As per claim 2, Burroughs teaches a method wherein: said data access request includes a search operation (i.e. *"This allows for increased efficiencies because queries can specify what partition to search for objects in and avoid searching large amounts of unneeded objects."* The preceding text clearly indicates that in order to submit queries a search operation is performed.)(Column 3, lines 18-21).

As per claim 3, Burroughs teaches a method wherein: said data access request includes adding new data to a data store (i.e. *"The next step 806 is to store the entity data for the owning entity in the appropriate row of the appropriate table if the data for the owning entity it has not been previous stored"* The preceding text clearly indicates that if the data for the owning entity has not been previous stored means that the entity data, which is new data, is stored in the data store, which is the appropriate table.)(Column 13, lines 42-44).

As per claim 4, Burroughs teaches a method wherein: said data access request includes accessing data in (i.e. *"FIG. 3 illustrates exemplary persistent data stored in Company, Employee and Supplies tables. FIG. 3 also illustrates the type of mapped objects that would come from the Company, Employee and Supplies tables. In particular, Company objects, Employee objects and Supplies objects. As a general rule, each table corresponds to a class of objects (i.e., a company class, an employee class and a supplies class) and each row in the table corresponds to an instantiated object from the corresponding class." "Storing partitioning information as part of primary key of object allows queries to performed against a partition without requiring the user to have any specific knowledge of the partitioning structure."* The preceding text clearly indicates that multiple data stores are company, employee, and supplies tables and data access request is a query.)(Column 9, lines 56-64; column 11, lines 46-48) both the relational databases (i.e. *"The schema mapping code defines the mapping between persistent storage in relational database tables and objects. As discussed above, this mapping preferably includes the defining and mapping of both partition key values and partition keys used to partition selected tables into two or more partitions. This mapping also defines the mapping of EntityOwningExtent (EOE) objects for each partition of each table."* The preceding text clearly indicates that one or more data stores of a set of data stores are inherent in a relational database tables and objects, one or more variables is the entityowningextent (EOE) objects.)(Column 13, lines 22-29)

Burroughs does not explicitly teach the method including the LDAP directory

Shigi teaches the method including the LDAP directory (i.e. *"A Repository Manager 16A translates the logical design used in the system into the structure and format of the repository. A Resource Manager 18A* ***stores and manages resources in a Resource Repository*** *54. A repository is typically, but does not have to be, a relational database." "In this example, the Core System is implemented as a Java application that can run on the following application platforms for the various functions required in the system: Operating System; Java Application Server;* ***Relational Database; and LDAP Directory Server****. At the bottommost layer, the Core System runs inside of a readily available operating system, such as Microsoft WINDOWS.TM. or freely available GNU/Linux on a server computer. The application code is written in the Java programming language and runs as a Web Application inside of a Web Application Server implementing the J2EE and Java Servlet 2.3 standards. Alternatively, a free application server, `Tomcat` is available from the Jakarta project of the Apache Foundation, while commercial application servers are sold by IBM, BEA Systems, and Macromedia Inc. among others. The Web Application Server is run inside of a Java Virtual-Machine (JVM) application that sits on top of the underlying computer operating system. JVM implementations are freely available from Sun Microsystems and IBM."* The preceding text clearly indicates that at least one relational database and an LDAP directory exists.)(Paragraphs [0040], [0088]).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Burroughs with the teachings of Shiigi to include a method including the LDAP directory with the motivation to integrate relational database systems with object oriented systems that provides for efficient partitioning of objects in the object oriented system. (Burroughs, column 3, lines 30-33).

As per claim 5, Burroughs teaches a method wherein: said data access request includes accessing data in only one data store (i.e. *"For example, if the new owned entity is a*

new employee object for a new company object, data for that new company object would need to first be stored in the a new row in the company table." The preceding text clearly indicates that data access request includes accessing data in only one data store, which is the company table.)(Column 13, lines 46-48).

As per claim 7, Burroughs teaches a method wherein: said plurality of data stores includes at least two relational databases (i.e. *"Relational databases have thus been commonly used to facilitate the efficient storage and retrieval of data."* The preceding text clearly indicates that there is exists more than one relational database that includes set of data stores, which are tables within a relational database.)(Column 2, lines 47-48).

As per claims 9, 22, 26, 30, 39 and 44, Burroughs teaches a method wherein: said step of sending includes translating said data access request to a format suitable for said relational database and communicating said translated data access request to said relational database (i.e. *"The schema mapper is responsible for converting the partitioning key into a set of columns in the table. Therefore, it must be left up to the schema mapper to provide this additional where clause."* The preceding text clearly indicates that a schema mapper translates the data access request to a format suitable for a relational database)(Column 12, lines 45-49).

As per claims 10, 31 and 45, Burroughs teaches a method further comprising: receiving a result from said relational database (i.e. *"In response to this request all of the data necessary to fill in the object is retrieved from the relational database 122."* The preceding text clearly indicates that retrieving a response is receiving a result from the relational database.)(Column 9, lines 20-21).

As per claims 11 and 32, Burroughs teaches a method further comprising: receiving a result from said relational database (i.e. *"In response to this request all of the data necessary to fill in the object is retrieved from the relational database 122."* The preceding text clearly indicates that retrieving a response is receiving a result from the relational database.)(Column 9, lines 20-21); and translating said result to said logical object class format (i.e. *"Thus, although the objects are ultimately stored in the legacy relational format, the storage format is transparent to the application program which can access the objects using object oriented techniques."* The preceding text clearly indicates that the results are stored in a logical class format, which is the storage format.)(Column 3, lines 7-11).

As per claims 12 and 33, Burroughs teaches a method wherein: said data access request is from an Identity System (i.e. *"Thus, what is a needed is a mechanism for integrating relational database systems with object oriented systems that provides for efficient partitioning of objects in the object oriented system."* The Identity System is an example of an object-oriented system and is an intended use of the prior art.)(Column 3, lines 31-34).

As per claims 13 and 34, Burroughs teaches a method wherein: said data access request is from an Identity and Access System (i.e. *"Thus, what is a needed is a mechanism for integrating relational database systems with object oriented systems that provides for efficient partitioning of objects in the object oriented system."* The Identity System is an example of an object-oriented system and is an intended use of the prior art.)(Column 3, lines 31-34).

As per claims 14 and 35, Burroughs teaches a method wherein: said plurality of data stores store identity information (i.e. *"Each class of the object includes one or more attributes that correspond to one or more columns in the corresponding table. In the illustrated example, the company object includes an ivCompCode attribute and an ivCompName attribute that corresponds to the CompCode and CompName column respectively."* The preceding text clearly indicates that ivCompCode, ivCompName are attributes, which are identity information that is stored in the data store, which are tables.)(Column 10, lines 3-8).

As per claim 17, Burroughs teaches a method wherein: said step of sending includes creating a custom filter for said data access request that is customized for said relational database to only include one or more variables mapped to said relational database (i.e. *"To facilitate the most flexible schema mapping and partitioning, the partition key value for an object can be defined by the developer to be an attribute of an owning object, a primary key field of an owning object, or constant defined for its class. When new owned objects are created the partition key values are retrieved from their specified location in the owning object to determine the partition in which the new owned object belongs. In the preferred embodiment the developer also specifies a corresponding column of fields in the owned objects table to store the partitioning key values. Preferably, the partitioning key values are stored as a subset of the primary key field for the owned object. Because the primary key field is used to uniquely identify the corresponding object, the partition information in the partition key value partially determines the unique identifier of the owned object."* The preceding text clearly indicates that partitioning key values inherently uses a customer filter to allow customized mapping of the relational database.)(Abstract).

4. Claims 15, 16, and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burroughs et al (U.S. Patent No. 6,341,289 and known hereinafter as

Burroughs) in view of Shiigi et al (U.S. Patent Pub. 2003/0014442 A1 and known hereinafter as Shiigi) and in further view of Bachmann et al (U.S. Patent No. 6,085,188 and known hereinafter as Bachmann).

As per claims 15 and 23, Burroughs and Shiigi do not explicitly teach a method wherein: said partition expressions are in LDAP filter format.

Bachmann teaches a method wherein: said partition expressions are in LDAP format (i.e. *"As seen in FIG. 5, the LDAP naming hierarchy includes a number of entries or nodes, with each entry or node represented by a unique entry identifier (EID). Thus, for example, the root node has an EID=1. Root has two (2) children, entry GB ("Great Britain") having an EID=2, and entry US ("United States") having an EID=3. Child node US itself has two (2) children, O=IBM (with EID=4) and O=Netscape (with EID=5). The remainder of the naming directory includes several additional entries at further sublevels."* The preceding text clearly indicates that LDAP format is the EID.)(Column 5, lines 13-21).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Burroughs with the teachings of Shiigi and further with the teachings of Bachmann to include a method wherein said partition expressions are in LDAP format with the motivation to integrate relational database systems with object oriented systems that provides for efficient partitioning of objects in the object oriented system. (Burroughs, column 3, lines 30-33).

As per claim16, Burroughs and Shiigi do not explicitly teach a method wherein: said first format is a logical object class format that is compatible with LDAP filter format.

Bachmann teaches a method wherein: said first format is a logical object class format that is compatible with LDAP filter format (i.e. *"In a preferred embodiment, each LDAP attribute that can be searched by the user is mapped to an attribute relation that consists of two columns: EID and normalized attribute value. As described above, each LDAP entry is assigned a unique identifier (EID). Based on the attribute syntax, the attributes are converted (or normalized) so that the invention can apply Structured Query Language (SQL) queries to the attribute values. For example, if the attribute syntax is case insensitive (CIS), the attribute value will be converted to all upper case and stored in an attribute table. The attribute table is used mainly for search operations to find the entries that match the filter criteria. The actual entry data is preferably stored in an ldap.sub.-- entry table, as will be described. Thus, the generated SQL queries use the attribute table to locate the entry EIDs that match the filter expression. Then, the EIDs are used to retrieve the entry data from the ldap.sub.-- entry table."*)(Column 6, lines 26-41).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Burroughs with the teachings of Shiigi and further with the teachings of Bachmann to include a method wherein said first format is a logical object class format that is compatible with LDAP filter format with the motivation to integrate relational database systems with object oriented systems that provides for efficient partitioning of objects in the object oriented system. (Burroughs, column 3, lines 30-33).

5. Claims 18, 19, 24, 27, 29, 36, 38, 40-43, and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burroughs et al (U.S. Patent No. 6,341,289 and known hereinafter as Burroughs) in view of Shiigi et al (U.S. Patent Pub. 2003/0014442

A1 and known hereinafter as Shiigi) and in further in view of Mullins (U.S. Patent No. 6,999,956).

As per claim 18, Burroughs and Shiigi do not explicitly teach a method wherein: said data access request includes a filter; said filter includes said one or more variables; and said using of said mappings to evaluate partition expressions for said data stores includes determining whether said filter overlaps with said partition expressions based on said mappings.

Mullins teaches a method wherein: said data access request includes a filter (i.e. *"This allows more developer control as to how datasets are exchanged, filtered and/or validated between a first data source and a second data source."* The preceding text clearly indicates that a data access request may include a filter, where data access request are datasets.)(Column 25, lines 31-34); said filter includes said one or more variables; and said using of said mappings to evaluate partition expressions for said data stores includes determining whether said filter overlaps with said partition expressions based on said mappings (i.e. *"There are distinctive advantages when using a dynamic mapping layer where a java object provides translation by mapping such objects of a first data source (relational or object database) and also mapping such object to an XML or other second format data source."* The preceding text clearly indicates that a filter is used to map between data stores, which are within a relational database and partition expressions which are java objects.)(Column 24, lines 27-34).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Burroughs with the teachings of Shiigi and further with the teachings of Mullins to include a method wherein: said data access request includes a filter; said filter includes said one or more variables; and said using of

said mappings to evaluate partition expressions for said data stores includes determining whether said filter overlaps with said partition expressions based on said mappings with the motivation to integrate relational database systems with object oriented systems that provides for efficient partitioning of objects in the object oriented system. (Burroughs, column 3, lines 30-33).

As per claims 19, 29, 36, 40 and 43, Burroughs and Shiigi do not explicitly teach a method wherein determining whether said filter overlaps with said partition expressions based on said mappings comprises: determining whether child sub-filters of said filter expression overlap with said partition expressions and combining results of said determination of whether child sub-filters overlap to determine whether said filter expression overlaps with said partition expressions, if said filter expression is a composite expression.

Mullins teaches a method wherein determining whether said filter overlaps with said partition expressions based on said mappings comprises: determining whether child sub-filters of said filter expression overlap with said partition expressions and combining results of said determination of whether child sub-filters overlap to determine whether said filter expression overlaps with said partition expressions, if said filter expression is a composite expression (i.e. *"There are distinctive advantages when using a dynamic mapping layer where a java object provides translation by mapping such objects of a first data source (relational or object database) and also mapping such object to an XML or other second format data source."* The preceding text clearly indicates that a filter is used to map between data stores, which

are within a relational database and partition expressions which are java objects.)(Column 24, lines 27-34).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Burroughs with the teachings of Shiigi and further with the teachings of Mullins to include a method wherein determining whether said filter overlaps with said partition expressions based on said mappings comprises: determining whether child sub-filters of said filter expression overlap with said partition expressions and combining results of said determination of whether child sub-filters overlap to determine whether said filter expression overlaps with said partition expressions, if said filter expression is a composite expression with the motivation to integrate relational database systems with object oriented systems that provides for efficient partitioning of objects in the object oriented system. (Burroughs, column 3, lines 30-33).

As per claims 24 27, 38, and 42 Burroughs and Shiigi do not explicitly teach a method wherein determining whether said filter overlaps with said partition expressions based on said mappings comprises: performing a partition compare function using said filter expression and a first partition expression to determine whether said filter expression overlaps with said first partition expression, if said filter expression and said first partition expression are both simple expressions; and performing said partition compare function by treating said filter expression as an input partition expression and treating said first partition expression as an input filter expression in order to determine

whether said filter expression overlaps said first partition expression, if said first partition expression is a composite expression.

Mullins teaches a method wherein determining whether said filter overlaps with said partition expressions based on said mappings comprises: performing a partition compare function using said filter expression and a first partition expression to determine whether said filter expression overlaps with said first partition expression, if said filter expression and said first partition expression are both simple expressions (i.e. *"FIG. 2 starts at position IA (Start/End) with a request step 2A from an object application (101 in FIG. 1), which calls the CocoBase DataSource Manager and Abstraction layer (hereafter CocoBase system) and delegates the database accessing to CocoBase. The logical loop that the CocoBase system performs begins in response to the request step (2A) and first performs the Extract Object Name step (4). The CocoBase system loop continues onward to the Evaluate Object Properties step (5) and its associated components where (i) object properties are evaluated (5), (ii) object to relational maps are accessed, created, or updated by 3B (and this may involve retrieving maps from the system DataStore 3C or from a separate Optional Maps Files Storage facility (1C)), and (iii) database commands necessary for data retrieval (including SQL strings) are generated at step 2B under the control of the Execution Status Monitor (1B), before the CocoBase system continues onward to the Obtain Data step (6). In step 6, the CocoBase system in coordination with components 1B, 2B and 3B, passes SQL strings or other DataBase Access Statements from 2B to the JDBC Driver (2C), which obtains data from the system DataStore (3C) and passes that data to the CocoBase system."*)(Column 14, lines 61-67; column 15, lines 1-14); and performing said partition compare function by treating said filter expression as an input partition expression and treating said first partition expression as an input filter expression in order to determine whether said filter expression overlaps said first partition expression, if said first partition expression is a composite expression (i.e. *"Then the CocoBase system places the obtained data in an object (Step 6A, Place Data in Object and sends the*

Data Object (Step 7A, Send Data Object) to the Requestor Object Application (back to Step 2A position) and the delegation to the CocoBase system by the Object Application ends (back to Step 1A).")(Column 15, lines 14-19).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Burroughs with the teachings of Shiigi and further with the teachings of Mullins to include a method wherein determining whether said filter overlaps with said partition expressions based on said mappings comprises: performing a partition compare function using said filter expression and a first partition expression to determine whether said filter expression overlaps with said first partition expression, if said filter expression and said first partition expression are both simple expressions; and performing said partition compare function by treating said filter expression as an input partition expression and treating said first partition expression as an input filter expression in order to determine whether said filter expression overlaps said first partition expression, if said first partition expression is a composite expression with the motivation to integrate relational database systems with object oriented systems that provides for efficient partitioning of objects in the object oriented system. (Burroughs, column 3, lines 30-33).

As per claims 37, 41, and 46, Burroughs and Shiigi do not explicitly teach a method wherein said step of providing comprises: removing terms not supported by a first data source from said information for said filter expression provided to said first data source.

Mullins teaches a method wherein said step of providing comprises: removing terms not supported by a first data source from said information for said filter expression provided to said first data source (i.e. *"Once the XML file is written, it can be edited using a standard text editor or XML editor, and can be modified to reflect map customization requirements. The following discussion relates to specific features of the generated XML file. The basic repository format has CBObject (CocoBase object class) definitions that reflect the select, insert, update, delete and call related map definitions for a CocoBase map. Each of those operations further consists of tables, fields and clauses that may exist to specify how the object is mapped to and from the data source."*)(Column 14, lines 6-15).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Burroughs with the teachings of Shiigi and further with the teachings of Mullins to include a method wherein said step of providing comprises: removing terms not supported by a first data source from said information for said filter expression provided to said first data source with the motivation to integrate relational database systems with object oriented systems that provides for efficient partitioning of objects in the object oriented system. (Burroughs, column 3, lines 30-33).

Allowable Subject Matter

6. Claim 20 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Argument

7. Applicant's arguments filed 21 June 2007 have been fully considered but they are not persuasive for the reasons set forth below.

Applicant argues:

"Neither references alone or in combination teaches or suggests determining one or more data stores of a plurality of data stores to service data access requests wherein the plurality of data stores comprises of at least one relational database and at least one LDAP directory."

The Examiner disagrees. The combination of Burroughs and Shiigi teaches determining one or more data stores of a plurality of data stores (i.e. "database tables and objects")(Column 13, lines 22-29) to service data access requests (i.e. "runtime requests") (Column 13, lines 22-29) wherein the plurality of data stores (i.e. "database tables and objects") comprises of at least one relational database and at least one LDAP directory (i.e. *"A Repository Manager 16A translates the logical design used in the system into the structure and format of the repository. A Resource Manager 18A* ***stores and manages resources in a Resource Repository*** *54. A repository is typically, but does not have to be, a relational database." "In this example, the Core System is implemented as a Java application that can run on the following application platforms for the various functions required in the system: Operating System; Java Application Server;* ***Relational Database; and LDAP Directory Server****. At the bottommost layer, the Core System runs inside of a readily available operating system, such as Microsoft WINDOWS.TM. or freely available GNU/Linux on a server computer. The application code is written in the Java programming language and runs as a Web Application inside of a Web Application Server implementing the J2EE and Java Servlet 2.3 standards.*

Alternatively, a free application server, `Tomcat` is available from the Jakarta project of the Apache Foundation, while commercial application servers are sold by IBM, BEA Systems, and Macromedia Inc. among others. The Web Application Server is run inside of a Java Virtual-Machine (JVM) application that sits on top of the underlying computer operating system. JVM implementations are freely available from Sun Microsystems and IBM." The preceding text clearly indicates that at least one relational database and an LDAP directory exists.)(Shiigi, paragraphs [0040], [0088]).

Hence, the Applicant's arguments do not distinguish over the claimed invention over the prior art of record.

Any other arguments by the applicant are either more limiting than the claimed language or completely irrelevant.

Conclusion

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Farhan M. Syed whose telephone number is 571-272-7191. The examiner can normally be reached on 8:30AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Gaffin can be reached on 571-272-4146. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

FMS

JEFFREY GAFFIN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100